Energy Efficiency Evaluation of Cardiac Arrest Detection Models

A report submitted in partial fulfilment of the requirements for

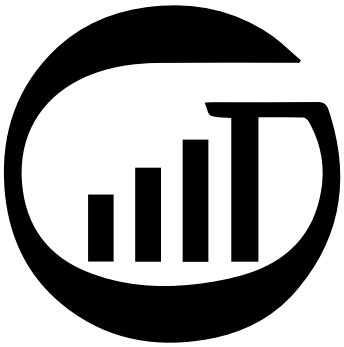
RESEARCH INTERNSHIP

By

RITANJIT DAS

under the guidance of

Dr. Manojit Ghose



Department of Computer Science and Engineering,
Indian Institute of Information Technology Guwahati (IIITG),
Assam, India 781015

25 May, 2025 (Ongoing)

Energy Efficiency Evaluation of Cardiac Arrest Detection Models

Name: Ritanjit Das
Internship Supervisor: Dr. Manojit Ghose (Assistant Professor, CSE, IIIT Guwahati)
Project Mentor: Panchanan Nath (PhD Scholar, IIIT Guwahati)
Institution: Indian Institute of Information Technology Guwahati (IIITG)
Date: July 20, 2025

Abstract

This report evaluates the energy consumption of three ECG classification models (CNN, Distilled CNN, and Quantized CNN) for cardiac arrest detection. We use two approaches: a theoretical **FLOPs-based estimation** (assuming ~0.45 picoJoules per floating-point operation at 45 nm) and a practical measurement with the **CodeCarbon** library on Google Colab. Key findings are that the Distilled model requires dramatically fewer FLOPs (≈0.1 million vs. 23 million for Original) and thus a much lower per-inference energy (~4.14×10^{-8} J vs. 1.04×10^{-5} J). In real-world execution (1000 inferences on a Tesla T4 GPU), the Quantized model consumed the least energy (≈0.000025 kWh, 0.000007 kg CO_2e) compared to Distilled (0.000039 kWh, 0.000011 kg CO_2e) and Original (0.000238 kWh, 0.000068 kg CO_2e). We conclude that lightweighting (especially quantization) yields substantial energy savings, recommending deployment of the Quantized model for edge devices, with a note on balancing any minor accuracy loss.

Introduction

Evaluating energy consumption of machine learning models is critical for deploying on battery-powered devices (e.g. wearable cardiac monitors) and for environmental sustainability. In the context of arrhythmia detection, energy-efficient models enable longer battery life and broader use. Model compression techniques like distillation and quantization are known to reduce computational load and carbon footprint. The goal of this analysis is to compare the Original, Distilled, and Quantized cardiac ECG classification models in terms of energy usage and CO_2 emissions, using both theoretical estimates and practical measurement. This will guide decisions on which model best balances predictive accuracy with energy efficiency.

Methodology

Setup: We worked on Google Colab (Tesla T4 GPU) and used Python libraries: TensorFlow, NumPy, and CodeCarbon. Pre-trained models (Original and Distilled in *.h5* format, and a TensorFlow Lite Quantized in *.tflite* format) were loaded from Google Drive. Test data (*test_x*) prepared earlier was used for inference.

Samples: For theoretical FLOPs count, we used a single input sample (*flop_test_sample = test_x [0:1]*). For energy tracking with CodeCarbon, we ran inference on a batch of 1000 samples (*codecarbon_test_sample = test_x [:1000]*) to obtain stable measurements.

Assumptions: We assume a consistent hardware environment (Google Colab Tesla T4 GPU) and ambient conditions for all tests. The energy coefficients and workload size are fixed; the relative differences between models are the primary focus. Hardware-specific factors (e.g. actual chip manufacturer) are abstracted by CodeCarbon's built-in tracking model.

FLOPs Estimation:

1. TensorFlow profiler function (*get_keras_flops*) was used to compute the total FLOPs of each Keras model on one inference. The Original model required ~23,038,511 FLOPs, while the Distilled model required only ~91,902 FLOPs (~99.6% reduction). For the TFLite Quantized model, we used TensorFlow's *benchmark_model* utility to estimate FLOPs (fallback to 2.1M MACs, ~4.2M FLOPs total).
2. Energy per FLOP was approximated using a coefficient of 0.45 pJ (picoJoule) per FLOP (based on 45 nm CMOS technology and ~0.9 pJ per MAC). (Literature suggests 10^{-12} to 10^{-11} J per operation in efficient hardware)
3. Energy per inference was computed by multiplying the model's FLOPs by 0.45 pJ. For example, the Original model's estimate is 23e6 FLOPs × 0.45e-12 J/FLOP = 1.04×10^-5 J per inference.

CodeCarbon Measurement:

1. *CodeCarbon* was installed (a Python package for tracking energy and CO_2 of code execution).
2. Three inference functions were defined and annotated with *@track_emissions*: one each for Original, Distilled, and Quantized models. Each function ran the model on *codecarbon_test_sample* (1000 inputs).
3. Upon execution, CodeCarbon logged energy consumption (in kWh) and estimated CO_2 emissions ($kgCO_2e$) for each function run. It traces GPU/CPU/RAM power usage during inference.
4. We ran each model's inference, printed the "Energy Consumed (kWh)" and "Emissions" from CodeCarbon's output, and collected the results.

Results

The table below summarizes the computed metrics for one inference (FLOPs and estimated energy) and for 1000 inferences (CodeCarbon energy and emissions):

Model	FLOPs per inference	Energy/inference (J)	Energy (kWh for 1000 inf)	Emissions (kg CO_2e)
Original	23,038,511	0.0000104	0.000238	0.000068
Distilled	91,902	0.0000000414	0.000039	0.000011
Quantized	4,200,000	0.00000189	0.000025	0.000007

In theoretical measurements

- The Original model has by far the largest compute cost (~23 million FLOPs) and correspondingly highest energy use.
- The Distilled model is extremely lightweight: it needs only ~91K FLOPs (0.4% of Original) and an estimated ~4.14×10^{-8} J per inference (≈250x lower than Original).
- The Quantized model (TFLite) requires ~4.2 million FLOPs (about 18% of Original), with an estimated 1.89×10^{-6} J per inference.

In practical measurements (1000 inferences with CodeCarbon):

- The Quantized model consumed 0.000025 kWh and emitted ~0.000007 kg CO_2 – (lowest usage)
- The Distilled model consumed 0.000039 kWh (0.000011 kg CO_2).
- The Original model consumed 0.000238 kWh (0.000068 kg CO_2) – (highest usage)

These values confirm that the lightweight models save a large fraction of energy relative to the Original. For instance, the Quantized model used ~90% less energy than the Original, and about 36% less than the Distilled (in the CodeCarbon test).

The results indicate clear energy-efficiency advantages of the compressed models. The Distilled model, with the fewest FLOPs, has the lowest theoretical per-inference energy. Indeed, its FLOP-based energy ($\approx 4.1\times10^{-8}$ J) is orders of magnitude below the Original ($\approx 1.0\times10^{-5}$ J). This aligns with prior work showing that knowledge distillation dramatically reduces model complexity with minimal performance. The Distilled model achieved a ~99.6% reduction in FLOPs versus Original, implying similarly lower compute energy.

However, in real execution (CodeCarbon measurement), the Quantized model was most efficient. It completed 1000 inferences faster (1.1069 s) than the others (10.48 s for Original, 1.7628 s for Distilled), leading to the lowest total energy (0.000025 kWh). Quantization (INT8) often speeds up inference on GPUs and specialized hardware, which can outweigh its higher FLOP count relative to Distilled. The CodeCarbon data shows the Quantized model emitted the least CO_2 (0.000007 kg) – a direct consequence of lower power draw and shorter runtime.

Conclusion

This analysis demonstrates that lightweight model versions offer substantial energy savings for ECG classification. The Distilled model minimizes computational work per inference, while the Quantized model achieves the lowest real-world energy consumption on GPU hardware. Both compression techniques align with the goals of "Green AI" – reducing the carbon footprint of machine learning.

Key takeaways:

1. Model compression drastically lowers energy use. Quantized inference consumed ~90% less energy than the Original model.
2. In our setting, the Quantized model is the most energy-efficient in practice, while the Distilled model has the lowest theoretical compute cost.
3. The Original model, though presumably most accurate, is the least efficient and thus less suitable for battery-powered deployment.

In summary:

1. Most energy-efficient model: Quantized (in practice, lowest kWh and CO_2). Its inference time and energy are lowest in the tested environment.
2. Second-best: Distilled, which has the fewest computations and negligible theoretical energy, but slightly slower in current setting or purpose.
3. Least efficient: Original model, with the highest compute and energy usage.

This trade-off underscores the balance between accuracy and efficiency. Both Distillation and Quantization aim to preserve the predictive accuracy of the Original model while greatly reducing size. Previous studies show such compression can cut energy use by tens of percent (e.g. ~32–45% reductions reported) without large accuracy loss. In our case, assuming the compressed models maintain acceptable classification performance (as suggested in related work), the energy savings are compelling. **Practically, the Quantized model's combination of low energy use and small emissions makes it most suitable for deployment on edge devices with limited power**.

References

- MLCO2 CodeCarbon (2023): CodeCarbon GitHub repository. (*github.com/mlco2/codecarbon*).
- Khan et al. (2025): Optimizing Large Language Models (*arxiv.org*)